A new ensemble method for outlier identification

Stamatios-Aggelos N. Alexandropoulos
Computational Intelligence Laboratory (CILab)
Department of Mathematics
University of Patras
GR-26110 Patras, Greece
Email: alekst@math.upatras.gr

Sotiris B. Kotsiantis
Computational Intelligence Laboratory (CILab)
Department of Mathematics
University of Patras
GR-26110 Patras, Greece
Email: sotos@math.upatras.gr

Violetta E. Piperigou
Division of Statistics, Probability and Operational Research
Department of Mathematics
University of Patras
GR-26110 Patras, Greece
Email: vpiperig@math.upatras.gr

Michael N. Vrahatis
Computational Intelligence Laboratory (CILab)
Department of Mathematics
University of Patras
GR-26110 Patras, Greece
Email: vrahatis@math.upatras.gr

***Abstract*—A vast number of factors influence the applicability of machine learning methods and the use of statistical models for a given task. The existence of outliers in a data set is a common issue that needs to be tackled. The identification of such values is a difficult, yet very useful project. In many cases errors or dissimilar values to the majority of the data are useless. Nevertheless, valuable information can be hidden in outliers' set. During the last years, although several models have been developed for outlier detection, there is always space for new, intelligent, more efficient and less time consuming techniques for this issue. In the present work we provide a new ensemble method for outlier detection. In order to test the proposed methodology, comparisons are made with widely used techniques for outlier detection. The results obtained indicate that our model is robust and quite competitive to the other methods.**

***Keywords*—machine learning, outlier detection, ensemble model**

I. Introduction

A common problem that is observed in data sets is the presence of *outliers* or *anomalies*. This problem is quite common and can significantly affect the performance of machine learning algorithms or statistical models [1]. The rapid growth over the last decade in the field of materials and technologies have made handling of big data possible. Consequently, big amounts of data can be used in the domain of Computer Science and thus, useful results can be drawn. Furthermore, new strategies and models belonging in the domain of Machine Learning and Data Mining make the reliable data necessary. This issue is important, in order to build accurate, intelligent and quick Computing Systems.

The terms *outlier* or *anomaly* or *exception* or *discordant* are often used for this type of data [2]. In most cases the first two of them are widely accepted and recognized. An important question that requires an answer is the following: "*What kind of data are the outliers?*" These data are frequently *errors* in recorded values or *exceptions* and thus, do not characterize the common functioning of the majority data. As a consequence, a really useful transaction is to perform an outlier removal technique in preparation to further proceeding. Although one might think that outliers are useless, this is not always true, since outliers can provide substantial information about existing *anomalies* in the data set. Therefore, the identification of outliers is a beneficial task due to the additional information that they can provide as with about a given data set.

The importance of a "clean" data set can be easily understood if the desired good generalization ability of a Machine Learning method is taken into account. If a data set is *complex* or *huge* or is having specific other problems, then the method under construction is possible to give poor results concerning unknown data. Generally, in the field of Machine Learning we are interested in predicting an outcome specified by some data [3] and many techniques used in Machine Learning can be seen as typical probabilistic methods. Machine Learning shares also similarities with Statistics as researchers care about good predictions. In addition, computational efficiency plays an important role in the construction of an algorithm [3]. Typically, the scope is either to provide some initial insights linked to an application field where minor a priori knowledge exists or, in addition, to be able to forecast quickly and accurately future observations.

In the literature *univariate* and *multivariate* outliers have been considered [4]. Univariate outliers concern a single feature in one dimensional space, while multivariate outliers concern many features in a multi-dimensional area. In order to tackle the problem of *outlier identification* or *outlier removal* a plethora of techniques have been developed. These techniques are categorized as [4]: (a) *Supervised*: When the data set is divided into two labeled sets, one as *normal* and a second one as *abnormal* and include the training of a classifier, (b) *Unsupervised*: When outliers are detected in an unlabeled test data set, assuming that the majority of the data in the set are normal and (c) *Semi-supervised*: When a method is constructed introducing normal behavior from a given normal training data set, and then examine the probability of a test example to be produced by the learned algorithm.

In the paper at hand a new methodology is proposed, in

978-1-7281-2791-0/20/$31.00

order to examine whether an instance is an outlier or not. To test our method, well-known and widely used outlier detection techniques have been applied for comparisons. The experimental results obtained strongly indicate that the provided scheme is robust and provides competitive results.

The rest of our paper is organized as follow: In Section II related works as well as well-known algorithms and techniques that tackles the problem of outlier detection are presented. In Section III our ensemble method and the proposed methodology are analyzed along with the experimental results. Finally, Section IV discusses the concluding remarks and some future research work.

II. Related works and Well-known techniques

As it has already been pointed out in the introduction of the current work, the issue of the outlier detection or/and elimination is a very important task, and it is related to many Data Mining projects [5], [6], [7]. Towards this goal many methods have been recently developed. We review the most reliable and well-known models for the outlier detection problem. For convenience these methods are categorized and presented into six classes.

A. Nearest neighbor based methods

The most well-known algorithms in this category are the k-*Nearest Neighbors (kNN) Detector* and the *AveKNN* [8]. In details, for an instance, the former algorithm evaluates the distance to its k-th nearest neighbors. This distance is the outcome result-score which is the mechanism to count the density. In the latter method, three *kNN* detectors can be used: (a) Firstly, the *largest*: In this detector the distance to the k-th neighbor is thought as the outlier measurement, (b) Secondly, we consider the *mean*: In this strategy the average of all *kNN* is used as the outlier score and (c) Finally, we consider the *median* detector: In this approach the median of the distances is used as the outlier outcome.

Recently, an unsupervised model named Histogram-Based Outlier Score (HBOS) has been developed [9]. In this approach the main idea is based on histograms as indicated by their name. In particular, the model considers self-determination of the features producing a very fast method in relation to other multivariate methods in terms of the cost of less accuracy.

In [10] a model that is based on the Local Outlier Factor (LOF) has been developed. This method identifies anomalies in multidimensional data sets. The above mentioned factor points out whether an element is an outlier or not. Thus, the algorithm evaluates the local deviation of density of a particular point in relation to its neighbors. Accurately, locality is specified by *kNN*, whose distance is used in order to evaluate the local density. Through this comparison of the local density of an example to the densities of its neighbors, one can detect cases that have a considerably lower density than their neighbors. These values are evaluated as anomalies.

B. Probabilistic methods

In [11] the authors have developed a model that it can be applied in high-dimensional data sets and differs from distance based approaches. Specifically, the so-called *Angle-Based Outlier Detection (ABOD)* model uses angles instead of distances. In general, the angle provides a more robust measure than the distance in the case of high-dimensional data. The authors have observed that the angle variance is more accurate and in order to speed up the overall process, a subset of data has been used for evaluation.

An outlier detection approach named *Stochastic Outlier Selection (SOS)* has been provided in [12]. It is a stochastic method and it is based on affinity in order to determine the relation between the data. Affinity accounts for the amount of similarity between two values. Thus, a data value is estimated as outlier if the affinity amount to all the other points is poor.

A combination of the *Mahalanobis Squared Distance (MSD)* and the *Minimum Covariance Determinant (MCD)* covariance estimator has been used in [13] in order to produce a method for anomaly detection. In particular, if a value is not included in any cluster, then it is considered as outlier. From the statistical point of view the estimator of the Minimum Distribution Coefficient can only be applied to data from the Gaussian distribution, but could also be satisfactorily used for data from a unimodal symmetric distribution.

C. Subspace based methods

The well-known and widely used *Principal Component Analysis (PCA)* has been used in [14] in order to detect anomalies in a given data set. In this approach, the eigenvectors of the data covariance matrix with high eigenvalues validate most of the variance in the data. Accordingly, the dimensionality is reduced and a hyperplane is created by k eigenvectors of the major components which capture most of the variance in the data. Anomalies that are quite dissimilar to the normal data values are measured both with their distance from the major and the minor components.

In [15] a *Subspace Outlier Detection (SOD)* method which tackles the outlier identification problem in different subclasses of a high dimensional space has been developed. Using this method is not possible to find an outlier in the starting data space. Thus, for each data, *SOD* searches the axis-parallel subspace in order to find out how much the object differs from the neighbors in this subspace.

D. Ensemble methods

A model that does not depend on distances or density measures have been given in [16]. Specifically, the provided method, is called *Isolation Forest (iForest)* and distinguishes outliers based on the idea of isolation. A fact worth noting is that *iForest* is capable to take advantage of sub-sampling. This algorithm can reach a small linear time complexity as well as a small memory-requirement. In addition, it can handle effectively two frequently observed phenomena, namely: the *swamping* and the *masking*.

An ensemble, unsupervised method which is based on local regions near a test instance, named *LSCP (Locally Selective Combination in Parallel Outlier Ensembles)* has been presented in [17]. Specifically, the local region is determined in order to be the collection of the nearest training data in randomly sampled feature subclasses which take place more

often than a determined threshold over multiple iterations. The usage of the local region has the consequence of a local pseudo ground truth is clarified and the Pearson correlation is intended between each main detector's training anomaly outcomes and the pseudo ground truth. Moreover, a histogram is created by the Pearson correlation outcomes. Finally, the concluding outcome is the average anomaly score of the selected qualified detectors.

In [18] a novel approach which detects outliers, combining the decision of multiple outlier algorithms for high-dimensional and noisy data sets have been proposed. As a result, diverse outlier scores have been observed. A *Feature Bagging (FB)* detector has been used as a meta estimator that adjusts an amount of base detectors on diverse sub-samples of the original data set and use averaging or other combination models in order to reach a better forecasting accuracy and to avoid the affect of over-fitting.

E. Classifier based methods

In [19] the authors have proposed a model, named *One-Class Support Vector Machine (OCSVM)*, that is similar to the well-known support vector method in order to detect the outliers of a data set. Thus, the option of a kernel is required as well as a scalar parameter. More often the choice of *RBF kernel* is preferred.

F. Clustering based methods

An introduction of a new approach named *Cluster-Based Local Outlier Factor (CBLOF)* has been presented in [20]. The model that used *CBLOF* has been called *FindCBLOF* method and the experimental results have been shown that the provided method is quite competitive in comparison to other clustering variations. In particular, the proposed model takes into consideration the original data set and the clustering algorithm. The outlier score is then delivered based on the size of the cluster the data reside in as well as the distance from the nearest large cluster.

III. Proposed methodology and experimental results

Ensemble methods are models that researchers increasingly trust. This is due to the fact that these methods can be empirically built and they combine the decisions of reliable methods in order to make an even more reliable and efficient final model. Numerous methods have been presented for the creation of such a set of classifiers [21]. In this section we analyze a new ensemble method for outlier identification. In the sequel, we provide the methodology that is adopted in our method. The proposed methodology, which constitutes a six-step strategy, is exhibited in the following Algorithm I.

It is worth mentioning that the proposed ensemble model can simply be parallelized using a detector per machine. Parallel and distributed computing is of great significance for practitioners, since when taking advantage of a parallel or a distributed implementation a methodology may: i) increase its speed, and ii) extend the range of applications where it can be used (i.e., the model can process more data).

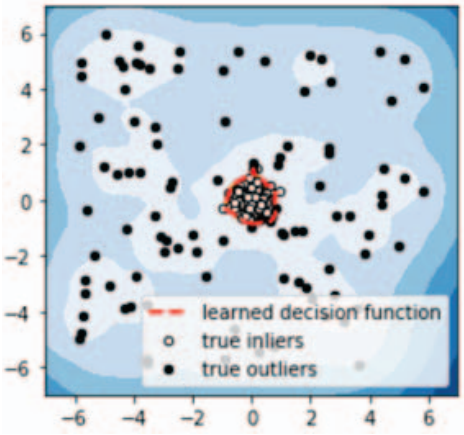
It is also worth noting that the time complexity of the selection of the best subset of detectors increases with respect to the number of base detectors that are used. From this point of view the heuristic rule to test the algorithms using a small subset of the training set decreases the computational complexity. The detectors that are initially used for building the ensemble are tested in a small subset of the training set and only the best three participate to the final decision of the ensemble model.

Algorithm I: ***Proposed methodology***

1 :	A sample of about 20% of the initial set of data is extracted.
2 :	The extracted data set is divided at random into three equal or almost equal parts.
3 :	Two of these three parts are used for training of detectors and the remaining data are used as the testing set.
4 :	The outcome of three tests are averaged.
5 :	The three algorithms that have obtained the best ROC performance are selected in order to build a strong final method.
6 :	The three best algorithms are executed on the whole initial set to produce the prediction model by averaging scores of all detectors.

For a simplified illustrative example the aforementioned methodology is exhibited in the Fig. 1 below.

Fig. 1. A simplified illustrative example



Consider an initial data set with $n = 400$ data points where the outlier fraction is $o_{fraction} = 0.25$. Thus, we have $n_{inliers} = 300$ points and $n_{outliers} = 100$ points. Next, as indicated at Step 1, 80 sample data points are randomly taken. Then, the extracted data set, according to Step 2, is divided into three parts with 27, 27 and 26 data points. At Step 3, two of the above mentioned sets are used as training sets, while the remaining one is used as the test set. Specifically, the black points in Fig. 1 illustrate the true outliers of the data, while the white ones denote the inliers. The outcomes of the three tests are averaged as it is introduced at Step 4. Next, the three efficient and effective learners (in our case, the *CBLOF*, the *IForest* and the *HBOS* were the three algorithms with the best ROC performance for the given example) are combined in order to built a stronger final classifier. Our methodology terminates at Step 6, where the best algorithms are executed on the whole initial data and the points that are included within

TABLE I. RECEIVER OPERATING CHARACTERISTICS (ROC) CURVES

Data	#Samples	# Dimensions	Outlier Perc	ABOD	CBLOF	FB	HBOS	IForest	KNN	LOF	MCD	OCSVM	PCA	SOS	SOD	AveKNN	LSCP	Proposed
arrhythmia	452	274	14.6018	0.7921	0.7892	0.7841	0.8379	0.8173	0.7917	0.7833	0.8317	0.7832	0.7861	0.6701	0.7528	0.7913	0.8062	0.8290
cardio	1831	21	9.6122	0.5694	0.8167	0.6223	0.8387	0.9186	0.7381	0.5790	0.8162	0.9328	0.9480	0.5351	0.6730	0.6739	0.9274	0.9361
glass	214	9	4.2056	0.8149	0.8857	0.9038	0.7703	0.7770	0.8828	0.9238	0.7996	0.5917	0.6396	0.7696	0.8378	0.8565	0.7914	0.9044
ionosphere	351	33	35.8974	0.9287	0.9004	0.8892	0.6128	0.8465	0.9274	0.8951	0.9544	0.8286	0.7932	0.7774	0.8858	0.9315	0.8225	0.9382
letter	1600	32	6.2500	0.8787	0.7864	0.8865	0.5750	0.6298	0.8771	0.8713	0.8010	0.6147	0.5378	0.8347	0.8941	0.9061	0.6950	0.8956
lympho	148	18	4.0541	0.9383	0.9944	0.9926	1.0000	1.0000	0.9833	0.9926	0.9377	0.9926	0.9963	0.7652	0.9435	0.9778	1.0000	1.0000
mnist	7603	100	9.2069	0.7897	0.8518	0.7137	0.5790	0.8062	0.8522	0.7097	0.8490	0.8595	0.8600	0.5410	0.6841	0.8406	0.8164	0.8547
musk	3062	166	3.1679	0.0631	1.0000	0.4400	1.0000	0.9995	0.7252	0.4377	0.9998	1.0000	0.9998	0.4776	0.8632	0.4054	0.9564	1.0000
optdigits	5216	64	2.8758	0.4673	0.5345	0.5120	0.8555	0.7185	0.3895	0.5249	0.3914	0.5226	0.5215	0.5077	0.4981	0.3765	0.6080	0.7273
pendigits	6870	16	2.2707	0.6945	0.8673	0.4655	0.9298	0.9605	0.7576	0.4850	0.8372	0.9350	0.9361	0.5096	0.7082	0.7390	0.9499	0.9489
pima	768	8	34.8958	0.6770	0.6517	0.5924	0.6827	0.6566	0.6988	0.6069	0.6686	0.5933	0.6107	0.5200	0.5983	0.6976	0.6644	0.6930
satellite	6435	36	31.6395	0.5769	0.7545	0.5666	0.7570	0.6928	0.6895	0.5646	0.8055	0.6636	0.5958	0.4783	0.6347	0.6739	0.6740	0.7723
satimage-2	5803	36	1.2235	0.8026	0.9999	0.4034	0.9916	0.9988	0.9479	0.3960	0.9959	0.9999	0.9943	0.5222	0.8003	0.9318	0.9956	0.9995
vertebral	240	6	12.5000	0.3891	0.4195	0.3990	0.3216	0.3878	0.3821	0.4010	0.3971	0.4534	0.4251	0.5312	0.4503	0.3739	0.3608	0.4783
vowels	1456	12	3.4341	0.9660	0.9276	0.9533	0.6862	0.7678	0.9795	0.9484	0.7566	0.7967	0.6226	0.7533	0.9030	0.9793	0.7242	0.9749
wbc	378	30	5.5556	0.8976	0.8847	0.9110	0.9575	0.9064	0.9201	0.9082	0.9017	0.9085	0.8860	0.6780	0.9254	0.9167	0.9019	0.9332

TABLE II. PRECISION-RECALL CURVE (PRC)

Data	#Samples	# Dimensions	Outlier Perc	ABOD	CBLOF	FB	HBOS	IForest	KNN	LOF	MCD	OCSVM	PCA	SOS	SOD	AveKNN	LSCP	Proposed
arrhythmia	452	274	14.6018	0.4410	0.4821	0.4560	0.5744	0.5617	0.4560	0.4554	0.5078	0.4821	0.4954	0.3615	0.4415	0.4693	0.5355	0.5572
cardio	1831	21	9.6122	0.2464	0.3932	0.1646	0.4576	0.4769	0.3348	0.1634	0.4134	0.4900	0.5816	0.1595	0.3293	0.3120	0.5031	0.5249
glass	214	9	4.2056	0.2778	0.2222	0.3889	0.0556	0.2222	0.2222	0.3889	0.0000	0.2222	0.0556	0.3889	0.2222	0.2222	0.0556	0.3889
ionosphere	351	33	35.8974	0.8492	0.8065	0.7205	0.4419	0.6401	0.8624	0.7203	0.8847	0.7135	0.5868	0.6663	0.7865	0.8779	0.6422	0.8750
letter	1600	32	6.2500	0.3924	0.2129	0.4515	0.1015	0.0991	0.3242	0.3833	0.1660	0.1562	0.0903	0.4504	0.4825	0.3872	0.1363	0.4615
lympho	148	18	4.0541	0.4167	0.9167	0.9167	1.0000	1.0000	0.9167	0.9167	0.3333	0.9167	0.9167	0.5833	0.7500	0.9167	1.0000	1.0000
mnist	7603	100	9.2069	0.3637	0.4062	0.3115	0.1233	0.3209	0.4139	0.3308	0.2929	0.3925	0.3825	0.1474	0.3025	0.4058	0.3384	0.4086
musk	3062	166	3.1679	0.0123	1.0000	0.1111	0.9753	0.9259	0.1605	0.1235	0.9630	1.0000	0.9630	0.0741	0.2840	0.0864	0.7408	0.9918
optdigits	5216	64	2.8758	0.0000	0.0000	0.0317	0.2173	0.0063	0.0000	0.0254	0.0000	0.0000	0.0000	0.0403	0.0000	0.0000	0.0063	0.0964
pendigits	6870	16	2.2707	0.0939	0.2289	0.0605	0.3368	0.3958	0.1240	0.0662	0.0994	0.3958	0.3647	0.0415	0.0771	0.1118	0.4003	0.3869
pima	768	8	34.8958	0.5314	0.4587	0.4285	0.5234	0.4810	0.5207	0.4284	0.5045	0.4243	0.4472	0.3374	0.4581	0.5197	0.4811	0.5252
satellite	6435	36	31.6395	0.3991	0.5872	0.4084	0.5716	0.5769	0.5054	0.4030	0.6866	0.5356	0.4781	0.2777	0.4595	0.4985	0.5521	0.6169
satimage-2	5803	36	1.2235	0.2455	0.9507	0.0604	0.7401	0.9277	0.3934	0.0675	0.6494	0.9507	0.8506	0.0533	0.2591	0.3782	0.8435	0.9430
vertebral	240	6	12.5000	0.0256	0.0513	0.0256	0.0000	0.0513	0.0256	0.0256	0.0000	0.0256	0.0000	0.1282	0.1025	0.0256	0.0000	0.0940
vowels	1456	12	3.4341	0.5938	0.2357	0.3309	0.0701	0.1909	0.5255	0.3121	0.0889	0.2626	0.1177	0.2422	0.4282	0.5441	0.1399	0.5545
wbc	378	30	5.5556	0.1786	0.3762	0.4238	0.5321	0.4238	0.4238	0.4238	0.3595	0.4238	0.3821	0.1083	0.3821	0.4238	0.3821	0.4599

TABLE III. RANKINGS OF THE ALGORITHMS USING THE FRIEDMAN TEST (USING ROC)

Rank	Algorithm
2.43750	Proposed
6.43750	CBLOF
6.96875	IForest
7.00000	KNN
7.40625	MCD
7.43750	HBOS
7.62500	OCSVM
7.71875	LSCP
7.81250	AveKNN
8.59375	PCA
9.31250	SOD
9.56250	ABOD
9.56250	FB
9.62500	LOF
12.5000	SOS

TABLE IV. POST-HOC BONFERRONI-DUNN (USING PROPOSED AS CONTROL METHOD - USING ROC)

Comparison	Statistic	Adjusted p-value	Result
Proposed vs SOS	6.36408	0.00000	H_0 is rejected
Proposed vs LOF	4.54577	0.00008	H_0 is rejected
Proposed vs ABOD	4.50625	0.00009	H_0 is rejected
Proposed vs FB	4.50625	0.00009	H_0 is rejected
Proposed vs SOD	4.34813	0.00019	H_0 is rejected
Proposed vs PCA	3.89355	0.00138	H_0 is rejected
Proposed vs AveKNN	3.39945	0.00945	H_0 is rejected
Proposed vs LSCP	3.34016	0.01172	H_0 is rejected
Proposed vs OCSVM	3.28086	0.01449	H_0 is rejected
Proposed vs HBOS	3.16228	0.02192	H_0 is rejected
Proposed vs MCD	3.14251	0.02345	H_0 is rejected
Proposed vs KNN	2.88558	0.05470	H_0 is accepted
Proposed vs IForest	2.86581	0.05823	H_0 is accepted
Proposed vs CBLOF	2.52982	0.15977	H_0 is accepted

TABLE V. RANKINGS OF THE ALGORITHMS USING THE FRIEDMAN TEST (USING PRC)

Rank	Algorithm
2.50000	Proposed
6.75000	IForest
6.87500	CBLOF
7.12500	OCSVM
7.34375	KNN
7.56250	HBOS
7.68750	AveKNN
7.78125	LSCP
9.00000	FB
9.06250	SOD
9.21875	PCA
9.28125	MCD
9.37500	ABOD
9.40625	LOF
11.03125	SOS

TABLE VI. POST-HOC BONFERRONI-DUNN (USING PROPOSED AS CONTROL METHOD - USING PRC)

Comparison	Statistic	Adjusted p-value	Result
Proposed vs SOS	5.39564	0.00000	H_0 is rejected
Proposed vs LOF	4.36790	0.00018	H_0 is rejected
Proposed vs ABOD	4.34813	0.00019	H_0 is rejected
Proposed vs MCD	4.28884	0.00025	H_0 is rejected
Proposed vs PCA	4.24931	0.00030	H_0 is rejected
Proposed vs SOD	4.15049	0.00046	H_0 is rejected
Proposed vs FB	4.11096	0.00055	H_0 is rejected
Proposed vs LSCP	3.34016	0.01172	H_0 is rejected
Proposed vs AveKNN	3.28086	0.01449	H_0 is rejected
Proposed vs HBOS	3.20181	0.01912	H_0 is rejected
Proposed vs KNN	3.06346	0.03063	H_0 is rejected
Proposed vs OCSVM	2.92511	0.04821	H_0 is rejected
Proposed vs CBLOF	2.76699	0.07921	H_0 is accepted
Proposed vs IForest	2.68794	0.10065	H_0 is accepted

the red dashed circles are the learned decision points that are identified as the inliers by the model.

In order to test the proposed method the *Receiver Operating Characteristics (ROC)* curves are used as a measure of the performance of the methods considered in this comparison. *ROC* curves typically characterize true positive and false negative rate on the Y and X axis respectively. This indicates that the top left corner of the plot is the best possible point — a false positive rate of zero, and a true positive rate of one. This is not feasible actually, but it does indicate that a bigger *Area Under the Curve (AUC)* is commonly better. The obtained results are exhibited in Table I.

In addition, statistical tests have been carried out in order to test the significance of the outcomes. In particular, the non-parametric *Friedman test* [22] has been conducted. According to this test, the null hypothesis (all methods exhibit the same performance with respect to ROC) is strongly rejected. Consequently, the statistical tests indicate that there are models whose performance difference was statistically significant to the others. In Table III the ranking of the algorithms using the Friedman test is presented. Furthermore, the results obtained by the *post-hoc Holm Bonferroni-Dunn* test [23] using the proposed method as a control method are exhibited in Table IV. It can be seen that the proposed method performance is better than the other methods.

Furthermore, it has been used the *Precision-Recall Curve (PRC)* that indicates the trade-off between precision and recall for different threshold. Since high area as possible under the curve indicates both high recall and high precision. If the precision is high, then we have a low false positive rate. On the other hand, if we have high recall, then low false negative rate is indicated. If both scores are high, which means that the detector is returning good outcomes (high precision), as well as returning a majority of all positive results (high recall). In Table II the results obtained for *PRC* are presented. The results of the statistical tests can be seen in Table V and VI.

In order to conduct this experimental study we used the well-known Python programming language. Specifically, the *PyOD* toolbox [24] has been used for the outlier detection algorithms. Moreover, the *STAC* web platform [25] has been used to conduct the comparisons and for obtaining the statistical tests.

IV. CONCLUSION

Significant progress has been made in relation to outlier detection methods. Many and varied methods have been developed in order to tackle the problem of outlier identification. The applications of the aforementioned problem are so many. For this reason the reliable handling of this problem is urgent and of high importance.

Ensemble methods seems to be an appropriate and reliable solution to several problems of machine learning. Thus, the construction of a new ensemble method for anomaly identification is presented in the work at hand. The new methodology is analyzed and the experimental results indicates that our method is robust and quite competitive. In a future correspondence, our aim is to improve the proposed method by embedding unsupervised feature selection.

ACKNOWLEDGMENT

Stamatios-Aggelos N. Alexandropoulos is co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme «Human Resources Development, Education and Lifelong Learning» in the context of the project "Strengthening Human Resources Research Potential via Doctorate Research" (MIS-5000432), implemented by the State Scholarships Foundation (IKY).

REFERENCES

[1] R. Domingues, M. Filippone, P. Michiardi and J. Zouaoui, A comparative evaluation of outlier detection algorithms: Experiments and analyses, Pattern Recognition, Vol. 74, pp. 406-421, 2018.

[2] V. Chandola, A. Banerjee and V. Kumar, Anomaly detection: A survey, ACM Computing Surveys, Vol. 41, no. 3, Article 15, 2009.

[3] I. H. Witten, E. Frank, M. A. Hall and C. J. Pal, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, 2016.

[4] V. Hodge and J. Austin, A survey of outlier detection methodologies, Artificial intelligence review, Vol. 22, no. 2, pp. 85-126, 2004.

[5] M. R. Smith and T. Martinez, Improving classification accuracy by identifying and removing instances that should be misclassified, In The 2011 International Joint Conference on Neural Networks, IEEE, 2011, pp. 2690-2697.

[6] K. Singh and S. Upadhyaya, Outlier detection: applications and techniques, International Journal of Computer Science Issues, Vol. 9, no. 1, pp. 307-323, 2012.

[7] S. Mishra and M. Chawla, M, A comparative study of local outlier factor algorithms for outliers detection in data streams, In Emerging Technologies in Data Mining and Information Security, Singapore: Springer, 2019, pp. 347-356.

[8] F. Angiulli and C. Pizzuti, Fast outlier detection in high dimensional spaces, In European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 2002, pp. 15-27.

[9] M. Goldstein and A. Dengel, Histogram-based outlier score (hbos): a fast unsupervised anomaly detection algorithm, KI-2012: Poster and Demo Track, 2012, pp. 59-63.

[10] M. M. Breunig, H. P., Ng, R. T. Kriegel, and J. Sander, LOF: identifying density-based local outliers, In ACM sigmod record, ACM, Vol. 29, No. 2, pp. 93-104, 2000.

[11] H. P. Kriegel, M. Schubert and A. Zimek, Angle-based outlier detection in high-dimensional data. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2008, pp. 444-452.

[12] J. H. M. Janssens, F. Huszar, E. O. Postma and H. J. van den Herik, Stochastic outlier selection. tech. rep., 2012.

[13] J. Hardin and D. M. Rocke, Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator, Computational Statistics & Data Analysis, Vol. 44, no. 4, pp. 625-638, 2004.

[14] M. L. Shyu, S. C. Chen, K. Sarinnapakorn and L. Chang, A novel anomaly detection scheme based on principal component classifier. In: Proc. of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03), Melbourne, Florida, 2003, pp. 172-179.

[15] H. P. Kriegel, P. Kroger, E. Schubert and A. Zimek, Outlier detection in axis-parallel subspaces of high dimensional data, In Pacific-Asia Conference on Knowledge Discovery and Data Mining, Berlin, Heidelberg: Springer, 2009, pp. 831-838.

[16] F. T. Liu, K. M. Ting and Z. H. Zhou, Isolation-based anomaly detection, ACM Transactions on Knowledge Discovery from Data, Vol. 6, no. 1, Article 3, 2012.

[17] Y. Zhao, Z. Nasrullah, M. K. Hryniewicki and Z. Li, LSCP: Locally selective combination in parallel outlier ensembles, In Proceedings of the 2019 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, 2019, pp. 585-593.

[18] A. Lazarevic and V. Kumar, Feature bagging for outlier detection, In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, ACM, 2005, pp. 157-166.

[19] J. C. Platt, J. Shawe-Taylor, A. J. Smola and R. C. Williamson, Estimating the support of a high-dimensional distribution, Technical Report MSR-T R-99Ű87, Microsoft Research (MSR), 1999.

[20] Z. He, X. Xu and S. Deng, Discovering cluster-based local outliers, Pattern Recognition Letters, Vol. 24, no. 9-10, pp. 1641-1650, 2003.

[21] T. G. Dietterich: Ensemble methods in machine learning. In: International workshop on multiple classifier systems. pp. 1-15. Springer, 2000.

[22] J. Hodges and E. L. Lehmann, Rank methods for combination of independent experiments in analysis of variance. The Annals of Mathematical Statistics Vol. 33, no. 2, pp. 482-497, 1962.

[23] O. J. Dunn, Multiple comparisons among means. Journal of the American Statistical Association, Vol. 56, Issue 293, pp. 52-64, 1961.

[24] Y. Zhao, Z. Nasrullah, and Z. Li, PyOD: A Python Toolbox for Scalable Outlier Detection. Journal of machine learning research, Vol. 20, no. 96, pp. 1-7, 2019.

[25] I. Rodríguez-Fdez, A. Canosa, M. Mucientes and A. Bugarín, STAC: a web platform for the comparison of algorithms using statistical tests, in: Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2015.